Union Theorems in Type-2 Computation

Chung-Chih Li

School of Information Technology
Illinois State University
Normal, IL 61790, USA

Abstract. The union theorem [12] indicates that, informally, almost all natural complexity classes at type-1 such as PTIME, PSAPCE, EXPTIME, EXPSPACE, and so on, fit the precise definition of complexity classes given by Hartmanis and Stearns in [3]. In other words, according to the theorem, the rigorous definition of complexity classes in terms of computable resource bounds is indeed broad enough to encompass most natural complexity classes. However, when we lift the computation to type-2 using oracle Turing machines, the union theorem doesn't hold without further strengthening some necessary conditions. In [8] we prove a non-union theorem under a less considered cost model known as unit-cost model. In this paper, we examine a more popular cost model known as answer-length-cost model and give a full treatment of this powerful theorem at type-2. We prove and disprove several nontrivial variations of the union theorem based on our framework.

1 Introduction

Let $\mathbf{N}$ be the set of natural numbers. By computable we mean Turing machine computable. A function is said to be recursive if it is total and computable. Let $\mathcal{R}$ denote the set of recursive functions. We use φ_e to denote the function computed by the e^{th} Turing machine. Thus, when we say the computation of φ_e, we refer it to the computation of the e^{th} Turing machine. Let Φ_e denote Blum's complexity measure [1] associated with the computation of φ_e. Clearly, there are infinitely many different Turing machines that compute the same function, φ_e, with different complexity. In their seminal paper [3], Hartmanis and Stearns give a precise definition of complexity classes as follows. For each $t \in \mathcal{R}$, the complexity class $\mathcal{C}(t)$ is defined by:

$$\mathcal{C}(t) = \left\{ f \in \mathcal{R} \,\middle|\, \exists e \left[\varphi_e = f \text{ and } \overset{\infty}{\forall} x \left(\Phi_e(x) \leq t(|x|) \right) \right] \right\}, \tag{1}$$

where $\overset{\infty}{\forall}\ x$ is understood as "*for all but finitely many*" and $|x|$ is the length of the bit representation of $x \in \mathbf{N}$. Despite the fact that the definition in (1) has provided a solid foundation for the study of complexity theory, we prefer to characterize computational complexity classes according to the properties of the resources bounds, not just to name a class by a single function t as shown in (1). For example, PTIME is a complexity class in which every problem can be solved by some Turing machine within a number of steps bounded by some polynomial. In other words, "being polynomial" is the property required for the time-bounds for all problems in PTIME. Therefore, we define,

$$\text{PTIME} \;=\; \left\{ f \;\middle|\; f \in \text{DTIME}(p) \text{ and } p \text{ is a polynomial} \right\},$$

where DTIME(p) follows the definition in (1). Thus, we could better understand PTIME as follows: PTIME $= \bigcup_{k\in\mathbf{N}} \text{DTIME}(n^k)$. Clearly, this union gives us a more intuitive idea about what PTIME is. However, it is not at all obvious that PTIME is indeed a complexity class under the formal definition in (1). Is there a recursive function that determines exactly the same complexity class, PTIME? The same question can be asked elsewhere, e.g., the big-O notation in algorithm analysis, which can be understood as $O(f) = \bigcup_{k\in\mathbf{N}} \text{DTIME}(k \cdot f)$. Is $O(f)$ a rigorously defined complexity class? The powerful union theorem provides a positive answer to this kind of questions we just asked. The theorem is proven by McCreight and Meyer [12], which is the first theorem in complexity theory proven by using a priority argument with finite injuries.

Theorem 1 (The Union Theorem [12]). *For any sequence of recursive functions $f_0, f_1, f_2, \ldots$ such that, $\lambda i, x.f_i(x)$ is recursive and, for all $i, x \in \mathbf{N}$, $f_i(x) \leq f_{i+1}(x)$, there is a recursive function g such that $\mathcal{C}(g) = \bigcup\limits_{i\in\mathbf{N}} \mathcal{C}(f_i)$.* ■

According to the union theorem, there is $g \in \mathcal{R}$ such that DTIME(g) = PTIME. Likewise, we can apply the theorem to $O(f)$, PSPACE, EXPTIME, etc. and claim that they are indeed complexity classes. Clearly, not any arbitrary collection of resource bounds satisfied the two conditions (i) and (ii) in the union theorem. For example, there is no such uniformly effective enumeration that can cover all computable bounds. Thus, we cannot use the union theorem to argue that the class of recursive functions is a complexity class. In fact, Blum [1] proves that given any $t \in \mathcal{R}$, there always exists a recursive function g such that, $g \notin \text{DTIME}(t)$. The simplicity of the two premises required in the union theorem above allows us to apply the theorem to most natural complexity classes. However, we shall argue that we cannot expect the same simplicity at type-2.

The most widely studied type-2 "complexity class" is $\mathbf{BFF}_2$ (Basic Feasible Functional at type-2). With Cook and Kapron's second-order polynomials [2, 4, 5], $\mathbf{BFF}_2$ seems to be a natural type-2 analog of PTIME. Is $\mathbf{BFF}_2$ a type-2 complexity class under some notion similar to (1)? Unfortunately, since there is no generally accepted machine model for type-2 complexity theory, we are not able to answer this question without a reasonable and workable framework to begin with. The framework must include the selection of computing formalism (i.e., an abstract machine such as the oracle Turing machine), the cost model for such machines, type-2 asymptotical notations, type-2 complexity measures, time bounds and a clocking scheme, and a precise definition of type-2 complexity classes. In the following section we shall give necessary terminology and notation in order to describe our union theorems at type-2. Details about our framework and concerns are discussed in [7–11].

2 Necessary Background for Type-2 Complexity Classes

We will try to keep our notation minimal due to the space constraints. Check [6] for A complete definitions of our notations and proofs. Let $\mathcal{F}$ and $\mathcal{T}$ denote the set of finite functions and total functions, respectively, over $\mathbf{N}$. With a fixed coding method for $\mathcal{F}$, we can assume that $\mathcal{F} \subset \mathbf{N}$ and treat any finite function as a natural number. Let $\sigma \in \mathcal{F}$. We use $\sigma \subset f$ to denote that f is an extension of σ. In [8] we define $\mathbf{T_2TB}$ (Type-2 Time Bounds) as a class of functions to be used as time bounds for clocking OTM (Oracle Turing Machines). OTM is considered as our formal computing device for type-2 computation. For convenience, we repeat the definition of $\mathbf{T_2TB}$ in the following.

Definition 1 (Type-2 Time Bounds) *Let $\beta : \mathcal{F} \times \mathbf{N} \to \mathbf{N}$. We say that:*

1. *β is nontrivial, if for every $(\sigma, a) \in \mathcal{F} \times \mathbf{N}$, $\beta(\sigma, a) \geq |a| + 1$;*
2. *β is bounded, if for every $(f, x) \in \mathcal{T} \times \mathbf{N}$, $\sigma \in \mathcal{F}$, and $\sigma \subset f$, we have $\beta(\sigma, x) \leq \lim_{\tau \to f} \beta(\tau, x)$;*
3. *β is convergent, if for every $(f, a) \in \mathcal{T} \times \mathbf{N}$, there exists $\sigma \in \mathcal{F}$ with $\sigma \subset f$ such that, for all τ with $\sigma \subseteq \tau$, we have $\beta(\sigma, a) = \beta(\tau, a)$; We use $\beta(\sigma, a) \downarrow$ to denote that β converges at (σ, a).*
4. *β is $\mathcal{F}$-monotone, if for every $a \in \mathbf{N}$ and $\sigma, \tau \in \mathcal{F}$ with $\sigma \subseteq \tau$, we have $\beta(\sigma, a) \leq \beta(\tau, a)$.*

If β is computable, nontrivial, bounded, and convergent, we say that β is a type-2 time bound. Moreover, if β is $\mathcal{F}$-monotone, we say that β is strong. ■

With an appropriate clocking scheme, a precise notion of type-2 complexity classes can be given. Recall from the classical complexity theory, the constructibility property imposed on resource bounds guarantees a basic hierarchy among classes (see [13], pages 68, 82-85). Intuitively, a time constructible function is an efficiently computable function that is large enough to be used as a time bound for some Turing machines to operate. The classical definition of constructibility is rather intuitive and straightforward. This, however, is not the case at type-2. Much of the difficulty is caused by the cost of making oracle queries and reading the answers returned from the oracle. In other words, making queries and taking answers may use up the resource granted by the resource bound. Note that, at type-1, the union theorem has no concern about constructibility. But at type-2, without a reasonable notion of constructibility, we can use a trivial counterexample to disprove the union theorem. Moreover, under the unit-cost OTM model, a rather strong non-union theorem can be proven (Theorem 5 in [8]) where the OTM is not required to read every bit of the oracle answers. In this paper, we will emphasize on the answer-length-cost model, which is a cost model that requires the OTM to read every bit of the answer returned from the OTM. However, we have to distinguish the two models in some definitions and theorems. If it is necessary, we use OTM^a (M_e^a) and OTM^u (M_e^u) to denote the OTM (with index e) under answer-length-cost model and unit-cost model, respectively. Similarly, for a result obtained based on a certain cost model, we use an “a” or “u” in superscription to indicate the concerned model.

In the following discussion, we will give some notions that are similar to the classical notion of time constructibility. However, we hesitate to consider any of these notions a type-2 analog of time-constructibility because at the present moment it is not clear how do these notions affect the time-hierarchy at type-2; they just serve the purpose of obtaining a reasonable union theorem at type-2. We first rule out those type-2 time bounds that are too small for any OTM to make queries. To successfully query $f(q)$, an OTM^u needs at least $|q|+1$ steps to place q onto the query tape, whereas an OTM^a needs another $|f(q)|+1$ steps to read the answer, $f(q)$. Let $\|\mathsf{dom}(\sigma)\| = \sum_{i\in\mathsf{dom}(\sigma)}(|i|+1)$. Therefore, $\|\mathsf{dom}(\sigma)\|$ is the minimum number of steps an OTM^u needs to query the entire domain of σ. We abuse the notation by $\|\sigma\| = \sum_{i\in\mathsf{dom}(\sigma)}(|i| + |\sigma(i)| + 2)$. Thus, $\|\sigma\|$ is the minimum number of steps for an OTM^a to query the entire domain of σ and read their answers. Let $M^u_{e,\beta}$ denote the machine obtained from clocking M^u_e with $\beta \in \mathbf{T}_2\mathbf{TB}$, and let $\varphi^u_{e,\beta}$ be the functional computed by $M^u_{e,\beta}$ (same as $M^u_{e,\beta}$ and $\varphi^u_{e,\beta}$). Moreover, let $\varphi^u_{e,\beta}(f,x)\Downarrow$ denote that the computation of $M^u_{e,\beta}(f,x)$ terminates and the value is the same as $\varphi^u_e(f,x)$. In other words, $\varphi^u_{e,\beta}(f,x)\Downarrow$ means $M^u_e(f,x)$ can finish its computation under β.

Definition 2 *Let $\beta \in \mathbf{T}_2\mathbf{TB}$ and $(\sigma,a) \in \mathcal{F}\times\mathbf{N}$.*

1. *We say that (σ,a) is **β-queriable**, if there is $M^u_{e,\beta}$, such that on every $(f,a) \in \mathcal{T}\times\mathbf{N}$ with $\sigma \subset f$, $M^u_{e,\beta}$ can successfully query $\mathsf{dom}(\sigma)$ in some order. We say that (σ,a) is **β-queriable witnessed** by OTM^u M^u_e.*
2. *We say that (σ,a) is **β-checkable**, if there is $M^u_{e,\beta}$, such that, on every $(f,x) \in \mathcal{T}\times\mathbf{N}$, $\varphi^u_{e,\beta}(f,x)\Downarrow$, and*

$$\varphi^u_{e,\beta}(f,x) = \begin{cases} 1 & \text{if } \sigma \subset f \text{ and } x = a; \\ 0 & \text{otherwise.} \end{cases}$$

*We say that (σ,a) is **β-checkable witnessed** by OTM^u M^u_e.* ∎

Since every $\beta \in \mathbf{T}_2\mathbf{TB}$ must be convergent, it is clear that that not every $(\sigma,x) \in \mathcal{F}\times\mathbf{N}$ is β-checkable or β-queriable. Suppose that (σ,a) is β-queriable witnessed by M^u_e. Although $M^u_{e,\beta}$ can gain budget by simply querying $\mathsf{dom}(\sigma)$, the budget however is based on information of σ. Thus, not for every $\tau \in \mathcal{F}$ with $\mathsf{dom}(\sigma) = \mathsf{dom}(\tau)$, (τ,a) is also β-queriable witnessed by some OTM^u. For a β-queriable (σ,a), β will provide enough budget for an OTM^u to print out $\mathsf{dom}(\sigma)$ in some order, but may not be enough for any OTM^a to do the same. If (σ,a) is β-checkable, then σ can be printed in some order by a β-clocked OTM^a on every (f,a) with $\sigma \subset f$. We further define two properties in the following with which the time bounds are more useful for our purposes.

Definition 3 *Let $\beta \in \mathbf{T}_2\mathbf{TB}$.*

1. *We say that β is **accessible** if and only if there is an OTM^u M^u_e such that, all minimal locking fragments of β are β-queriable witnessed by M^u_e.*
2. *We say that β is **useful** if and only if there is an OTM^u M^u_e such that, all minimal locking fragments of β are β-checkable witnessed by M^u_e.* ∎

For example, $\beta(\sigma, a) = |a| + |\sigma(a)| + 1$ is both accessible and useful, since it allows some OTM on input (f, a) to check value of $f(0)$. the value. Clearly, if β is useful, then it is also accessible. The reason we want β to be useful is as follows. Suppose OTM M_e can be computed under β. If β is useful, then we can patch e on some finitely many (τ, a) under the same budget provided by β as long as (τ, a) is not a locking fragment of β. We will see why we need this later. We say that β is locking detectable if there is a computable function to decide whether β will converge on (σ, x). A locking detectable β is not necessarily useful or accessible. Also, if $\beta \in \mathbf{T}_2\mathbf{TB}$ is accessible, then β is locking detectable. One can easily verify the following two properties: (1) For every $a \in \mathbf{N}$, $(\emptyset, a)$ is β-checkable. (2) For every $(\sigma, a) \in \mathcal{F} \times \mathbf{N}$, if (σ, a) is β-checkable, then $\beta(\sigma, a) \geq \|\sigma\| + |a| + 1$.

Let $\beta_1 \leq \beta_2$ denote that, for every $(\sigma, x) \in \mathcal{F} \times \mathbf{N}$, $\beta_1(\sigma, x) \leq \beta_2(\sigma, x)$. We now define the properties of a sequence of type-2 time bounds required in the union theorems.

Definition 4 *Let $\langle \beta_i \rangle$ denote a sequence of type-2 time bounds $\beta_0, \beta_1, \beta_2, \ldots$.*

1. *We say that $\langle \beta_i \rangle$ is **uniform** if and only if $\lambda i, \sigma, x.\beta_i(\sigma, x)$ is recursive.*
2. *We say that $\langle \beta_i \rangle$ is **ascending** if and only if, for all $i \in \mathbf{N}$, $\beta_i \leq \beta_{i+1}$.*
3. *We say that $\langle \beta_i \rangle$ is **useful** if and only if, for all $i \in \mathbf{N}$, β_i is useful.*
4. *We say that $\langle \beta_i \rangle$ is **convergent** if and only if, for every $(f, x) \in \mathcal{T} \times \mathbf{N}$, there is a $\sigma \subset f$ such that, $\beta_i(\sigma, x) \downarrow$ for every $i \in \mathbf{N}$.*
5. *We say that $\langle \beta_i \rangle$ is **uniformly convergent** if and only if, for every $n \in \mathbf{N}$ and $(\sigma, x) \in \mathcal{F} \times \mathbf{N}$, if $\beta_n(\sigma, x) \downarrow$, then for all $i \in \mathbf{N}$, $\beta_i(\sigma, x) \downarrow$.*
6. *We say that $\langle \beta_i \rangle$ is **strongly convergent** if and only if $\langle \beta_i \rangle$ is uniformly convergent and there is a recursive locking detector for β_0.* ■

Let $\langle \beta_i \rangle$ be strongly convergent and let ℓ be a locking detector for β_0. By definition, $\langle \beta_i \rangle$ is uniformly convergent. Thus, we can use ℓ to detect the convergence of the entire sequence. That is,

$$[\ell(\sigma, x) = 1] \implies \forall i \in \mathbf{N}[\beta_i(\sigma, x) \downarrow],$$

and, for all $(f, x) \in \mathcal{T} \times \mathbf{N}$, $\lim_{\sigma \to f} \ell(\sigma, x) = 1$.

Examples: For every $i \in \mathbf{N}$ and $(\sigma, x) \in \mathcal{F} \times \mathbf{N}$, define

$$\alpha_i(\sigma, x) = \begin{cases} \sigma(x) + |x|^{i+1} + 1 & \text{if } x \in \mathsf{dom}(\sigma); \\ |x|^{i+1} + 1 & \text{otherwise.} \end{cases}$$

$$\beta_i(\sigma, x) = \begin{cases} \sigma(x+i) + |x|^{i+1} + 1 & \text{if } (x+i) \in \mathsf{dom}(\sigma); \\ |x|^{i+1} + 1 & \text{otherwise.} \end{cases}$$

One can see that $\langle \alpha_i \rangle$ and $\langle \beta_i \rangle$ above are two sequences of type-2 time bounds. Clearly, $\langle \alpha_i \rangle$ is *uniform*, *ascending*, and *strongly convergent*, while $\langle \beta_i \rangle$ is *uniform*, but neither *ascending* nor *convergent*. Moreover, all type-2 time bounds in $\langle \alpha_i \rangle$ and $\langle \beta_i \rangle$ are useful. ■

Let F_β denote the limit functional determined by $\beta \in \mathbf{T}_2\mathbf{TB}$. That is, for every $(f, x) \in \mathcal{T} \times \mathbf{N}$, $F_\beta(f, x) = \lim_{\sigma \to f} \beta(\sigma, x)$.

Lemma 1. *Given uniform and ascending $\langle \beta_i \rangle$, if there exists a total continuous functional $H : \mathcal{T} \times \mathbf{N} \to \mathbf{N}$ such that, for every $i \in \mathbf{N}$, $F_{\beta_i} \leq H$, then $\langle \beta_i \rangle$ is convergent.* ■

Given any $\beta \in \mathbf{T}_2\mathbf{TB}$, define $\langle \beta_i \rangle$ as, for each $i \in \mathbf{N}$, let $\beta_i = i\beta$. It is clear that such $\langle \beta_i \rangle$ is a counterexample of the inverse of Lemma 1. Referring to the discussing in [7], for any two continuous $F, G : \mathcal{T} \times \mathbf{N} \to \mathbf{N}$, if the set $\{(f, x) \mid F(f, x) > G(f, x)\}$ is compact in $\mathbb{T}(F, G)$, we say that F is almost everywhere less than G, denoted as $F \leq_{\mathbf{2}}^{*} G$. If we relax $F_{\beta_i} \leq H$ in Lemma 1 to $F_{\beta_i} \leq_{\mathbf{2}}^{*} H$, then we have the following lemma which is stronger than the inverse of Lemma 1 in the sense that we do not require $\langle \beta_i \rangle$ to be convergent.

Lemma 2. *For any uniform and ascending $\langle \beta_i \rangle$, there is a total continuous functional $H : \mathcal{T} \times \mathbf{N} \to \mathbf{N}$ such that, for every $i \in \mathbf{N}$, $F_{\beta_i} \leq_{\mathbf{2}}^{*} H$.* ■

Note that the functional H in Lemma 2 is not necessarily computable unless we can effectively determine when does each β_i converge. If we can, then H is computable since $\langle \beta_i \rangle$ is uniform and, for every $(f, x) \in \mathcal{T} \times \mathbf{N}$, the minimal locking fragment (τ, x) and $F_{\beta_{(x + \|\tau\|)}}(f, x)$ can be effectively obtained.

3 Non-union Theorems

Let $\mathbf{C}(\beta)$ denote the type-2 complexity class determined by $\beta \in \mathbf{T}_2\mathbf{TB}$ [8]. Similarly, let $\mathbf{C}(\langle \beta_i \rangle)$ denote the union class $\bigcup_{i \in \mathbf{N}} \mathbf{C}(\beta_i)$. According to Theorem 2 in [8], if $\langle \beta_i \rangle$ is ascending, then, for every $i \in \mathbf{N}$, $\mathbf{C}(\beta_i) \subseteq \mathbf{C}(\beta_{i+1})$. Clearly, if $\langle \beta_i \rangle$ is strongly convergent with a locking detector ℓ, then each β_i is a strong type-2 time bound because each β_i can share the same locking detector ℓ. The strong convergence of $\langle \beta_i \rangle$ is strong property that turns out to be one of the necessary hypotheses in our type-2 analog of the union theorem. The following theorem indicated that $\mathbf{BFF}_2$ can be described by some $\langle \beta_i \rangle$. The proof uses some results in Cook and Kapron's [2, 4, 5].

Theorem 1 *There is a uniform and ascending $\langle \beta_i \rangle$ such that, $\mathbf{C}(\langle \beta_i \rangle) = \mathbf{BFF}_2$.* ■

The theorem above implies that there is a programming system for $\mathbf{BFF}_2$. Similar to PTIME, $\mathbf{BFF}_2$ can be viewed as a union of complexity classes where each is determined by a second order polynomial. However, we will see later that $\mathbf{BFF}_2$ is not a type-2 complexity class determined by any $\beta \in \mathbf{T}_2\mathbf{TB}$. We first observe that, for any $\langle \beta_i \rangle$ such that $\mathbf{C}(\langle \beta_i \rangle) = \mathbf{BFF}_2$, $\langle \beta_i \rangle$ is not convergent. This is easy to see since the depth of a second-order polynomial can be arbitrarily deep. Thus, any locking fragment will not be enough for some second-order polynomial with deeper depth to compute.

Just as with the type-1 theory, in general, the union of two arbitrary complexity classes is not always a complexity class. We will see in the next section that some conditions are needed in order to obtain a type-2 union theorem.

Theorem 2 (Weak Type-2 Non-union Theorem) *There exist* $\beta_1 \in \mathbf{T_2TB}$ *and* $\beta_2 \in \mathbf{T_2TB}$ *such that,* $\forall \alpha \in \mathbf{T_2TB}, \mathbf{C}(\alpha) \neq \mathbf{C}(\beta_1) \cup \mathbf{C}(\beta_2)$. ■

Let $\mathbf{C}^a(\beta)$ denotes the complexity class determined by β under the answer-length-cost model, and $\mathbf{C}^u(\beta)$ the complexity class under the unit-cost model. In contexts where the difference between the two models is of no importance, we then simply use $\mathbf{C}(\beta)$.

Theorem 3 (Type-2 Non-Union Theorem) *There is a uniform, ascending, useful, and convergent* $\langle \beta_i \rangle$*, such that* $\mathbf{C}^a(\langle \beta_i \rangle)$ *is not a type-2 complexity class.* ■

These negative results (non-union theorems) help us to find and justify our rather strong hypotheses for obtaining a type-2 union theorem. For example, convergence is a rather strong hypothesis, but the theorem above shows that it is not sufficient to have a union theorem. Thus, we have to further strengthen the hypothesis by including *uniform convergence*. Similarly, if we drop the *usefulness* in the hypotheses, then we can modify the proof of Theorem 3 and have the following negative result.

Corollary 1 *There is a uniform, ascending, and uniformly convergent* $\langle \beta_i \rangle$*, such that* $\mathbf{C}^a(\langle \beta_i \rangle)$ *is not a type-2 complexity class.* ■

Thus, the *usefulness* of $\langle \beta_i \rangle$ should be added as a necessary condition in our union theorem. However, it is unclear that usefulness together with uniform convergence are sufficient to obtain a type-2 union theorem.

Conjecture 1 *There is a uniform, ascending, useful, and uniformly convergent* $\langle \beta_i \rangle$*, such that* $\mathbf{C}^a(\langle \beta_i \rangle)$ *is not a type-2 complexity class.* ■

The following two lemmas are straightforward. We omit the proof.

Lemma 3. *Let* $\langle \beta_i \rangle$ *be useful. If there is an* $\alpha \in \mathbf{T_2TB}$ *such that* $\mathbf{C}^a(\langle \beta_i \rangle) = \mathbf{C}^a(\alpha)$*, then* $\langle \beta_i \rangle$ *is convergent.* ■

Lemma 4. *Let* $\langle \beta_i \rangle$ *be useful. If there is an* $\alpha \in \mathbf{T_2TB}$ *such that* $\mathbf{C}^a(\langle \beta_i \rangle) \subseteq \mathbf{C}^a(\alpha)$*, then* $\langle \beta_i \rangle$ *is convergent.* ■

If we allow $\langle \beta_i \rangle$ to be not useful, then Lemma 3, can be disproved by constructing a trivial $\langle \beta_i \rangle$. For example, let $\mathbf{C}^a(\beta_0) = \mathbf{C}^a(\beta_1) = \cdots$ where each β_i delays its convergence until an inaccessible point is reached. Thus, no OTM^a clocked by any β_i can query the inaccessible point. In such a way, each β_i in the sequence determines the same complexity class and hence $\mathbf{C}^a(\beta_0) = \mathbf{C}^a(\langle \beta_i \rangle)$ but the convergence of $\langle \beta_i \rangle$ breaks if we choose a different inaccessible point for each β_i to converge. Based on the discussion in this section, we have the following theorem as our conclusion.

Theorem 4 *There is no $\beta \in \mathbf{T_2TB}$ such that, $\mathbf{C}(\beta) = \mathbf{BFF}_2$.* ■

Using Lemma 4 we can further prove that, there is no $\beta \in \mathbf{T_2TB}$ such that, $\mathbf{BFF}_2 \subseteq \mathbf{C}^a(\beta)$. These negative non-union results imply that a straightforward type-2 analog of the Union Theorem does not exist. In the next section we show how to strengthen the hypotheses in order to have a type-2 Union Theorem under answer-length-cost model.

4 Union Theorems

According to Lemma 3, the convergence of $\langle \beta_i \rangle$ is a necessary condition for $\mathbf{C}(\langle \beta_i \rangle)$ to be a complexity class. However, Theorem 3 states that convergence together with uniformity, ascendancy, and usefulness are not sufficient to obtain a union theorem. Strong convergence turns out to be one of the necessary conditions as indicated in the following theorem. We use a priority argument with finite injuries to the theorem.

Theorem 5 (Type-2 Union Theorem) *Suppose that $\langle \beta_i \rangle$ is (i) uniform, (ii) ascending, (iii) useful, and (iv) strongly convergent. Then, there is an $\alpha \in \mathbf{T_2TB}$ such that, $\mathbf{C}^a(\alpha) = \mathbf{C}^a(\langle \beta_i \rangle)$.* ■

Both uniform and strong convergence are very strong conditions in the sense that, for every $(f, x) \in \mathcal{T} \times \mathbf{N}$, every β_i has to refer to the same fragment of f. At the moment, we do not see any reasonable way to get rid of this requirement of convergence. Here we discuss an unsuccessful try. We observe that the sample $\langle \beta_i \rangle$ constructed in the proof of the Type-2 Non-Union Theorem (Theorem 3) is not bounded, i.e., $\lim_{i\to\infty} F_{\beta_i}(f, x) = \infty$. We may ask, if $\langle \beta_i \rangle$ is bounded by some continuous functional, can we have a union theorem without requiring $\langle \beta_i \rangle$ to be uniformly convergent? The next corollary gives a negative result.

Corollary 2 *There exist a continuous functional $F : \mathcal{T} \times \mathbf{N} \to \mathbf{N}$ and a uniform, ascending, and useful $\langle \beta_i \rangle$ such that, for every $i \in \mathbf{N}$, $F_{\beta_i} \leq F$, and $\mathbf{C}^a(\langle \beta_i \rangle)$ is not a type-2 complexity class.* ■

Note that if $\langle \beta_i \rangle$ is bounded by a total continuous functional, then, by Lemma 1, $\langle \beta_i \rangle$ is convergent but not necessarily uniformly convergent.

Recall that a *strong* type-2 time bound is an $\mathcal{F}$-monotone one, i.e., for every $\sigma, \tau \in \mathcal{F}$ and $a \in \mathbf{N}$, $\sigma \subseteq \tau \Rightarrow \beta(\sigma, a) \leq \beta(\tau, a)$. We say that $\langle \beta_i \rangle$ is strong if and only if every β_i in $\langle \beta_i \rangle$ is $\mathcal{F}$-monotone. Computations clocked with such kind of time bounds have an intuitive advantage that the budget provided by the clock will never shrink during the courses of the computations. Thus, we may want the type-2 time bound α constructed in the proof of the type-2 Union Theorem to be strong. However, we are strongly skeptical about this. We have the following conjecture.

Conjecture 2 *There is a uniform, ascending, and strong $\langle \beta_i \rangle$ such that, if there is $\alpha \in \mathbf{T_2TB}$ such that $\mathbf{C}^a(\alpha) = \mathbf{C}^a(\langle \beta_i \rangle)$, then α is not strong.* ■

The Type-2 big-O Notation: The big-O notation is a key tool in algorithm analysis. A natural type-2 analog of the big-O notation can be defined as follows.

Definition 5 (Type-2 big-O Notation) *Given $\beta \in \mathbf{T_2TB}$, define*

$$\mathbf{O}(\beta) = \left\{ \varphi_e \;\middle|\; \varphi_e \in \mathbf{C}^a(c\beta + d) \text{ for some } c, d \in \mathbf{N} \right\}. \qquad \blacksquare$$

In fact, one of our primary motivations to have a type-2 union theorem is to examine whether $\mathbf{O}(\beta)$ is a well-defined type-2 complexity class. In our opinion, if the conditions in the our union theorem do not rule out $\mathbf{O}(\beta)$ to be a type-2 complexity class, we should consider the conditions reasonable, no matter how strong they are. Clearly, if the β is locking detectable, the the sequence β_i defined in $\mathbf{O}(\beta)$ is *strongly convergent*. Thus, by Theorem 5, we can prove the following corollary:

Corollary 3 *Let $\beta \in \mathbf{T_2TB}$. If β is locking detectable and useful, then there is an $\alpha \in \mathbf{T_2TB}$ such that $\mathbf{C}^a(\alpha) = \mathbf{O}(\beta)$.* $\blacksquare$

Note that, although we have Theorem 9 in [8] asserting that there is an effective operator Θ_L such that, $\Theta_L(\beta)$ is locking detectable and equivalent to β, but

$$[\mathbf{C}^a(\beta) = \mathbf{C}^a(\Theta_L(\beta))] \not\Rightarrow [\mathbf{C}^a(i\beta + i) = \mathbf{C}^a(i\Theta_L(\beta) + i)].$$

On the other hand, if we define $\beta_i = \Theta_L(i\beta + i)$, the strong convergence of $\langle \beta_i \rangle$ may not hold. This is because, if $i \neq j$, the inaccessible points of β_i and β_j are different. Thus, locking detectability of β is required in Corollary 3. We can easily prove the following two addition corollaries.

Corollary 4 *Let $\alpha, \beta \in \mathbf{T_2TB}$. If α and β are locking detectable and useful, then $\mathbf{O}(\alpha + \beta)$ is a type-2 complexity class.* $\blacksquare$

The following corollary states that we can drop the less significant term in the big-O notation. We omit the proof since it is straightforward.

Corollary 5 *Let $\alpha, \beta \in \mathbf{T_2TB}$. Suppose that both α and β are locking detectable and useful. If $\alpha \leq^* \beta$, then $\mathbf{O}(\alpha + \beta) = \mathbf{O}(\beta)$.* $\blacksquare$

5 Conclusion

For decades type-2 complexity theory using a machine model remains an untouched territory. This paper is added to a series of our previous ones devoted to building up this theory from scratch. As the framework becomes clearer due to our specific clocking scheme for OTM and the precise definition of type-2 complexity classes, we decided to push the theory further by proving a union theorem. Based on the theorem, as its type-1 counterpart, we can characterize some intuitive complexity classes in a precise way. Unfortunately, the most familiar $\mathbf{BFF}_2$ fails to pass the test, i.e., it is not a type-2 complexity class under our definition. This result on the one hand indicates that our framework may not be broad enough to encompass this intuitive type-2 complexity class. On the other hand, it may provide another legitimate reason to argue that $\mathbf{BFF}_2$ is not precise enough for further investigation on a theoretical base. The hindsight of our investigation in this paper may be that, we give a type-2 analog of the big-O notation and, according to the union theorem we proved, we can argue that it is a well-defined type-2 complexity class under our framework.

References

1. Manuel Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14(2):322–336, 1967.
2. Stephen A. Cook and Bruce M. Kapron. Characterization of the basic feasible functions of finite type. *Proceedings of the 30th Annual IEEE Symposium on the Foundations of Computer Science*, pages 154–159, 1989.
3. J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transitions of the American Mathematics Society*, pages 285–306, May 1965.
4. Bruce M. Kapron. Feasible computation in higher types. Ph.d. dissertation, University of Toronto, 1991.
5. Bruce M. Kapron and Stephen A. Cook. A new characterization of type 2 feasibility. *SIAM Journal on Computing*, 25:117–132, 1996.
6. Chung-Chih Li. Type-2 union theorems. `http://www.itk.ilstu.edu/faculty/chungli/mypapers/T2Unions.pdf`.
7. Chung-Chih Li. Asymptotic behaviors of type-2 algorithms and induced Baire topologies. In *Proceedings of the Third International Conference on Theoretical Computer Science*, pages 471–484, Toulouse, France, August 2004.
8. Chung-Chih Li. Clocking type-2 computation in the unit cost model. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *Proceedings of Computability in Europe, CiE 2006: Logical Approaches to Computational Barriers, CSR 7-2006*, pages 182–192, Swansea, UK, 2006.
9. Chung-Chih Li. Speed-up theorems in type-2 computation. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Proceedings of Computability in Europe, CiE 2007: Computation and Logic in the Real World*, pages 478–487, Siena, Italy, June 2007. Springer, LNCS 4497.
10. Chung-Chih Li. Query-optimal oracle Turing machines for type-2 computations. In *Proceedings of Computability in Europe, CiE 2008: Logic and Theory of Algorithms*, pages 293–303, Athens, Greece, June 2008.
11. Chung-Chih Li and James S. Royer. On type-2 complexity classes: Preliminary report. In *Proceedings of the Third International Workshop on Implicit Computational Complexity*, pages 123–138, Aarhus, Denmark, May 2001.
12. E. McCreight and A. R. Meyer. Classes of computable functions defined by bounds on computation. *Proceedings of the First ACM Symposium on the Theory of Computing*, pages 79–88, 1969.
13. Piergiorgio Odifreddi. *Classical Recursion Theory, Volume II*, volume 143 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishing, North-Holland, Amsterdam, 1999.